Pitfall of Google Tri-Grams Word Similarity Measure

Linda Wong Lin Juan, Bong Chih How, Johari Abdullah and Lee Nung Kiong
Faculty Computer Science and Information Technology, Universiti Malaysia Sarawak, Sarawak, Malaysia.
15020282@siswa.unimas.my

***Abstract*—This paper describes and examines Google Trigram word similarity based on Google n-gram dataset. Google Tri-grams Measure (GTM) is an unsupervised similarity measurement technique. The paper investigates GTM's word similarity measure which is the state-of-the art of the measure and we eventually reveal its pitfall. We test the word similarity with MC-30 word pair dataset and compare the result against the other word similarity measures. After evaluation, GTM word similarity measures is found significantly fall behind other word similarity measure. The pitfall of GTM word similarity is detailed and proved with evidences.**

***Index Terms*—Google Tri-grams; Pitfalls, Sentence Similarity, Text Similarity; Trigrams; Unsupervised; Word Similarity.**

I. INTRODUCTION

Text is composed of words and phrases. The two measures commonly used to gauge if two given text are similar are text similarity and text relatedness. Text similarity quantifies closeness of two texts. On the other hand, text relatedness is the degree of how two texts relate to each other. Theoretically, text relatedness is a function of word relatedness. Text relatedness measures are methods to quantify the relatedness of two texts while text similarity measures are methods that are used to identify how similar the texts to each other. According to Mihalcea Rada in guidebook of social science [1], there is an obvious relatedness between two phrases like *"We own a pet"* and "*I love animals*", even though they are obviously dissimilar. Text similarity and relatedness are two of the important area in the field of natural language processing and they are widely applied in real life like, detecting plagiarism [2], automatic question answering [3] that return candidate answers by evaluating textual data and information retrieval [4] as in searching for related articles based on the keywords like Google and Yahoo search engines.

To date, text similarity is computed by using word and phrase similarity. TrWP [5] is an unsupervised text similarity approach using both word and phrase similarity. It is a Bag-of-Word-and-Phrase (BoWP) approach where phrase-pair (unigram vs bi-gram or bi-gram vs bi-gram) are used to computes the text similarity. It adopts Sum-Ratio (product of sum and ratio between minimum and maximum of two numbers) to capture the strength of association between two overlapping Google n-grams based on the statistics in the Google n-gram dataset of overlapping n-grams associated with the two compared texts[5].

There is no lack of literatures since researchers like Landauer [6], Mihalcea [7], Li et. al [8], and Lin[9] wo have produced various text similarity measures. Well-known works like LSA[6] uses Singular Value Decomposition (SVD) to analyse the statistical relationships among words to find the semantic representation of words in a reduced dimensional space. To derive similarity, corresponding word vectors are computed of its cosine angle to obtain the text similarity. On the other hand, Li et al. [8] proposed a method that computes text similarity based on corpus statistics and syntactic information. The approach has also considered sequence of words of a text as it carries useful information and specific meaning. Liu [10] proposed a novel approach to compute short text similarity by considering semantic information, word order and the contribution of different parts of speech in a sentence. The overall sentence similarity is derived from a weighted combination of the distance between sub sequences.

In 2012, Islam [11] has reported that their proposed text similarity--Google Tri-grams Measure (GTM)--has outperformed many well-performed text similarities. The state-of-the-art of GTM measure is Google Tri-grams word similarity measure. Hence in this paper, we intend to detail how GTM word similarity works and at the same time, to highlight the pitfall of the measure. Lastly, we will present some evidences to verify the pitfall.

II. GOOGLE TRI-GRAMS WORD SIMILARITY MEASURE (GTM)

Google Trigrams Similarity Measure (GTSM) is a distributional method that uses a Google n-gram corpus dataset to find the inherent properties of similarity between texts. In general, GTSM has two main components: trigram word similarity and text similarity. Trigram The word similarity component is to derive word-word similarity which is the fundamental component that is required to derive the sentence similarity. The word-word scores are aggregated to deliver a score to represent text similarity. In this paper, we examine GTM word similarity.

The word similarity in GTM is derived through Google n-grams's tri-grams dataset. It takes into consideration all the tri-grams that begins and ends with the given pair of words regardless of their order. In additional, the most frequent unigram of each word is used to normalize the mean frequency of the tri-grams. The algorithm of the word similarity is described in detail in the following.

Given two words, w_a and w_b,

Step 1: First, obtain the sum of unigram frequency from Google unigram dataset, which is represented as F_{max}.

Step 2: Obtain the frequency of unigram w_a as $f(w_a)$, and w_b as $f(w_b)$ from Google unigram dataset.

Step 3: Between the unigram frequency of w_a and unigram frequency of w_b, choose the frequency of

the unigrams with minimum frequency as $min(f(w_a), f(w_b))$.

Step 4: Obtain the sum of the frequency of tri-grams that begins with w_a, ends with w_b as $f(w_a w_i w_b)$,

Step 5: Obtain the sum of the frequency of tri-grams that begins with w_b, end with w_a as $f(w_b w_i w_a)$.

Step 6: The information obtained from step 1 to step 5 is used to compute the word similarity which is defined as:

$$Sim\ (w_a, w_b) = \frac{\log \frac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times {F_{max}}^2}{f(w_a) \times f(w_b) \times\ min(f(w_a), f(w_b))}}{-2 \times \log \frac{min(f(w_a), f(w_b))}{F_{max}}} \quad (1)$$

In order to make sure that the $Sim\ (w_a, w_b)$ is always a positive number, there are three conditions as shown below.

$$Sim\ (w_a, w_b) = \begin{cases} \dfrac{\log \frac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times {F_{max}}^2}{f(w_a) \times f(w_b) \times\ min(f(w_a), f(w_b))}}{-2 \times \log \frac{min(f(w_a), f(w_b))}{F_{max}}} & if\ \frac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times {F_{max}}^2}{f(w_a) \times f(w_b) \times\ min(f(w_a), f(w_b))} \geq 1 \\ \dfrac{\log 1.01}{-2 \times \log \frac{min(f(w_a), f(w_b))}{F_{max}}} & if\ \frac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times {F_{max}}^2}{f(w_a) \times f(w_b) \times\ min(f(w_a), f(w_b))} \leq 1 \\ 0 & if\ \frac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times {F_{max}}^2}{f(w_a) \times f(w_b) \times\ min(f(w_a), f(w_b))} = 0 \end{cases} \quad (2)$$

The word similarity is computed based on the equation 2 referring to the condition, $\frac{\frac{1}{2}\left(\sum_{i=1}^{n_1} f(w_a w_i w_b) + \sum_{i=1}^{n_2} f(w_b w_i w_a)\right) \times {F_{max}}^2}{f(w_a) \times f(w_b) \times\ min(f(w_a), f(w_b))}$ which is calculated from the information collected from step 1 to step 5.

III. Walkthrough Of Google Tri-Grams Word Similarity Measure

In the following, we take an example from MC-30 to illustrate the steps to compute the word similarity score with GTM. Given two words, w_a="*car*" and w_b = "*automobile*":

Step 1: Obtain sum of all unigram frequency from unigram frequency which is F_{max}=605345293012.

Step 2: Obtain frequency of unigram "*car*" as $f(car)$ =107671676, and "*automobile"* as $f(automobile)$ = 4614763

Step 3: Compare the frequency of $f(car)$ and $f(automobile)$ to get minimum unigram frequency. Therefore, min $(f(car), f(automobile))$ = 4614763

Step 4: Obtain sum of frequency of tri-grams that begins with "*car*" and ends with "*automobile*", $\sum_{i=1}^{n_1} f(\text{"car"} w_i \text{"}automobile\text{"})$ = 56263.

Step 5: Obtain sum of frequency of tri-grams that begins with "*automobile*" and ends with "*car*", $\sum_{i=1}^{n_2} f(\text{"}automobile\text{"} w_i \text{"}car\text{"})$ = 114642.

Step 6: The information obtained from step 1 to step 5 is substituted into GTM word similarity equation (Equation 1).

$$Sim\ (car, automobile) = \frac{\log \frac{\frac{1}{2}(56263 + 114642) \times 605345293012^2}{107671676 \times 4614763 \times\ 4614763}}{-2 \times \log \frac{4614763}{605345293012^2}} \quad (3)$$

Therefore, $Sim\ (car, automobile) = 0.70$.

IV. Pitfall In Calculating The Word Similarity

The word-word similarity in GTM is computed by calculating the co-occurrence of compared words appears in Google's tri-grams. When the total of the occurrence is zero, the compared word yielded zero score. Therefore, an occurrence is required in order to secure a score more than 0. In our experiment, for example the walkthrough example, the word pair "*worked*" and "*CPU*" is does not co-occur in trigram, returning 0-word similarity whilst "*CPU*" and "*keeps*" doesn't seems to be similar but the word similarity score is 0.617 since the frequency of co-occurrence is high.

V. The Efficacy Of GTM On Measuring Word Similarity

In the following, we intend to evaluate GTM's word similarity against Li [19], Jiang and Conrath [7], Lin [8], Wu and Palmer [12], and Resnik [6] word similarity measures through Miller and Charles word pairs (MC30) [10]. MC30 is a dataset introduced by G.A. Miller and W.G. Charles. This dataset is commonly used to evaluate the accuracy of the word similarity measures by correlating the word similarity scores against human annotation scored. The annotation is mean of scores given by human judges scaled between 1 (less similar) to 4 (very similar). However, the output from

investigated sentence similarity measures ranged between 0 to 1. Therefore, the human annotation is normalized by using total number of scale which is 4 in order to obtain score of same space which is ranged 0 to 1 for comparison. The Pearson's correlation coefficient ranged between 0 to 1 is used to compare how much the sentence similarity measure correlates to the human annotated. The results of correlation between compared sentence similarity measures to human annotated score are recorded in Table 1.

Table 1
Results of word similarity measures against MC30 human annotation.

Word Similarity Measures	Correlation (r)
Li et al. [8]	0.891
Jiang and Conrath [12]	0.865
Lin [13]	0.834
Wu and Palmer [14]	0.803
Resnik [6]	0.795
GTM [11]	0.551

From the table, we can see that GTM's correlation score against human annotation is the lowest while the other compared approaches ranged the scores between 0.795 and 0.891 which a lot better than GTM word similarity measure. We inferred that the revealed pitfall of the similarity measure affects the performance of overall word similarity. The difference in correlation score between GTM and Li et al. [8] which has the highest score is significant, which is 0.34. The difference between Resnik which scores the 2nd lowest to GTM is also significant which is 0.244. On the whole, performance of GTM is not so ideal as compared to other word similarity measures. Therefore, we further investigate the GTM score of each word pairs by comparing to the normalized human annotation scores by the experts to prove that the pitfall has actually impact the performance of GTM word similarity.

The result of GTM word similarity score of each word pairs is shown in Table 2.

As we can observe from Table 2, the word similarity scores yielded by GTM are zero among 10 word pairs out of total of 30 pairs. Take an example, the word pair "*Asylum*" and "*Madhouse*" (No. 6) has recorded 0.90 from human judges but scored zero with GTM's word similarity. Upon examining the Google trigram dataset, there is no occurrence found in the dataset. From google trigram dataset, trigram that begins with "*Asylum*" has a total of 36417; trigram that begins with "*Madhouse*" has a total of 1923. Trigram that ends with "*Asylum*" has a frequency of 2962100; trigram that ends with "*Madhouse*" has a frequency of 76331. However, the frequency of trigrams that begins with "*Asylum*" and ends with "*Madhouse*" is 0; the frequency of trigrams that begins with "*Madhouse*" and ends with "*Asylum*" is 0. From the results, we can infer that words with similar meaning are less likely to co-occur.

In contrast, the word pair "*coast*" and *"forest*" (No. 25) has recorded 0.11 which is least similarity by human judgement but GTM's word similarity has a score of 0.60. If we examine google tri-gram dataset, trigram that begins with "*coast*" has a total of 271524; trigram that begins with "*forest*" has a total of 284445. Trigram that ends with "*coast*" has a frequency of 29033075; trigram that ends with "*forest*" has a frequency of 24111750. However, the frequency of trigrams that begins with "*coast*" and ends with "*forest*" is 4634; the frequency of trigrams that begins with "*Madhouse*" and ends with "*Asylum*" is 720. From the results, we can infer that word pair that has high co-occurrence in tri-gram is not necessarily similar.

Table 2
Results of GTM word similarity against human annotation from MC-30 dataset.

No.	Word Pairs	Normalized Human Annotation	GTM Word Similarity
1	Car-Automobile	0.98	0.70
2	Gem-Jewel	0.96	0.70
3	Journey-Voyage	0.96	0.60
4	Boy-Lad	0.94	0.68
5	Coast-Shore	0.93	0.65
6	Asylum-Madhouse	0.90	0.00
7	Magician-Wizard	0.88	0.74
8	Midday-Noon	0.86	0.67
9	Furnace-Stove	0.78	0.73
10	Food-Fruit	0.77	0.65
11	Bird-Cock	0.76	0.45
12	Bird-Crane	0.74	0.61
13	Tool-Implement	0.74	0.57
14	Brother-Monk	0.71	0.60
15	Lad-Brother	0.42	0.00
16	Crane-Implement	0.42	0.00
17	Journey-Car	0.29	0.53
18	Monk-Oracle	0.28	0.00
19	Cemetery-Woodland	0.24	0.65
20	Food-Rooster	0.22	0.00
21	Coast-Hill	0.22	0.60
22	Forest-Graveyard	0.21	0.59
23	Shore-Woodland	0.16	0.51
24	Monk-Slave	0.14	0.00
25	Coast-Forest	0.11	0.60
26	Lad-Wizard	0.11	0.00
27	Chord-Smile	0.03	0.00
28	Glass-Magician	0.03	0.53
29	Rooster-Voyage	0.02	0.00
30	Noon-String	0.02	0.00

VI. Conclusion And Future Work

In this paper, we have examined and discussed the pitfall of the word similarity of GTM. A. Islam [3] reported GTM similarity measure outperformed other sentence similarity measures. After evaluation of the word similarity measure, we discovered GTM scores the lowest correlation among the other replicated word similarity measures. We discovered one short-coming in GTM word similarity. This is because it depends heavily on frequency of co-occurrence of compared word in tri-grams dataset to secure a word similarity score higher than 0. It also proved in previous section that word pair that co-occur a lot in tri-gram dataset does not seems to be similar. As proved in the previous section, words of high similarity do not necessarily occur in trigram. GTM word similarity is proved to be zero when the trigram of the word pairs has zero frequency from the corpus.

For future work, we would like to investigate and evaluate GTM sentence similarity to discover the reason that outperformed other sentence similarity as reported in A. Islam's research [11] since it GTM word similarity's performance fall back behind other word similarity measures.

Acknowledgement

I would like to thank Universiti Malaysia Sarawak

(UNIMAS) and KPM who funded this research study through the grant FRGS/ICT07(04),1203,2014(04).

REFERENCE

[1] G. Ignatow and R. Mihalcea, *Text Mining: A Guidebook for the Social Sciences*, Sage Publications, 2016.

[2] M. J. Wise, "YAP3: Improved detection of similarities in computer program and other texts," *ACM SIGCSE Bulletin*, vol. 28, no. 1, pp. 130-134,1996.

[3] G. Liu, Z. Lu, T. Hao, and W. Liu, "Automatic Short Text Annotation for Question Answering System," *International Conference on Web Information Systems and Technologies. Springer, Berlin, Heidelberg*, 2010.

[4] G. Salton, J. Allan, and C. Buckley, "Approaches to passage retrieval in full text information systems," *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 1993.

[5] M. R. H. Rakib, A. Islam, and E. E. Milios. "TrWP: Text Relatedness using Word and Phrase Relatedness," In *SemEval@ NAACL-HLT*, pp. 90-95, 2015.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391-407, Sep. 1990.

[7] R. Mihalcea, C. Corley, and C. Strapparava. "Corpus-based and knowledge-based measures of text semantic similarity". In Proceedings of AAAI'06, 2006.

[8] Y. Li, Z. A. Bandar and D. McLean, "An approach for Measuring Semantic Similarity between Words Using Multiple Information Sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871-882, 2003.

[9] D. Lin, "An information-theoretic definition of similarity," In *Proc. of the 15th ICML*, pp. 296–304, 1998.

[10] X. Liu, Y. Zhou, R. Zheng, "Sentence similarity based on dynamic time warping," In *Semantic Computing, 2007 (ICSC 2007), International Conference on*, pp. 250-256, 2007. IEEE.

[11] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discov. from Data*, vol. 2, no. 2, pp. 10:1–10:25, Jul. 2008

[12] J. Jiang and D. Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy," In *Proc. of the International Conference on Research in Computational Linguistics ROCLING X*, 1998.

[13] D. Lin. "Automatic retrieval and clustering of similar words," In *Proc. of the 17th COLING*, pp. 768–774, 1998.

[14] Z. Wu and M. Palmer, "Verb semantics and lexical selection," In *Proc. of the 32nd annual Meeting of the Association for Computational Linguistics*, pp. 133 –138, New Mexico, USA, 1994.